

Student Name: Ali A. Hussain

Student id: 2003 sect #: 1 serial #:

6424

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

#### QUIZZ#4: Chapter 16 Prolog

\*\*\*\*\*

QUESTION: Define a Prolog predicate `range(I, K, L)` that creates a list containing all integers within a given range. It takes two integers `I` and `K` such that  $I \leq K$  and returns a list `L` containing all consecutive integers from `I` to `K`.

?- `range(4, 9, L)`.

`L = [4, 5, 6, 7, 8, 9]`

Yes

?- `range(67, 71, Y)`.

`Y = [67, 68, 69, 70, 71]`

Yes

?- `range(7, 1, Lst)`.

No

ANSWER:

$range(I, K, L) :- I \leq K, range(I, K, [I|L]),$

$Y = I + 1, I \text{ is } I + 1, (I, K, [I, L]).$

Student Name: Sayed Abbas Hashim Student id: 20054536 sect #: 1 serial #:

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

QUIZZ#4: Chapter 16\_Prolog

\*\*\*\*\*

QUESTION: Define a Prolog predicate **range(I, K, L)** that creates a list containing all integers within a given range. It takes two integers **I** and **K** such that  $I \leq K$  and returns a list **L** containing all consecutive integers from **I** to **K**.

?- range(4, 9, L) .

L = [4, 5, 6, 7, 8, 9]

Yes

?- range(67, 71, Y) .

Y = [67, 68, 69, 70, 71]

Yes

?- range(7, 1, Lst) .

No

ANSWER:

range(K, K, [K]).

range(I, K, L) :- ~~range(I, K, L)~~

~~range(I, K, L)~~ I < K,

~~K is K + 1~~ I<sub>2</sub> is I + 1,

range(I<sub>2</sub>, K, [I<sub>2</sub> | L]).

Student Name: mohammed h  
University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

\*\*\*\*\*QUIZZ#4: Chapter 16\_Prolog\*\*\*\*\*

containing the first element of each list in `l`. Each element of `l` is a nested list.

```
?- sublist_head([ [a,b], [b,c], [c,d] ], L).
L = [a,b,c]
```

*Handwritten notes:* "To us" with an arrow pointing to the list `[a,b,c]` in the output. "out list" with an arrow pointing to the list `[a,b,c]` in the output.

$\therefore \text{sub}(x) = \text{sub}(y) \iff x = y$   
 $\text{sub}(x) = \text{sub}(y) \iff (x, y) \in R$   
 $\text{sub}(x, y) = -$

Subtree-head ( $[1] / [1]$ )  $\rightarrow$  خاتمة

subInfo - head ([[]] | L<sub>low</sub>), ~~L~~ L<sub>low</sub> X :- ←

sublud - had (Lan, L) ~~cut~~. 20%, ignore ~~unwilling~~

subst. head ( $[CH | -] [L IN]^{out}$ ),  $[H | L out]$  :-

subtask - week (LIN, 1000)

Head of the  
first list  
is attached to  
Lout

★ First  
last  
contains Head  
8 tail so  
it must take  
it head &  
put in L out.

Club  
75  
Lout  
Hend 11



Student Name: Sayed Abbas Hushim

Stud id: 2004538 Sect#: 1 #: 21

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

QUIZZ#4: Chapter 16\_Prolog

\*\*\*\*\*

**QUESTION:** Write Prolog rules (predicates) named **rem** that takes 2 parameters: a list of numbers and a value D and produces a new list NOT containing the given value D.

?- rem([3, 2.4, 1.25, 9] , 2.4, U).  
U = [3, 1.25, 9]

**ANSWER:**

rem(~~[0, 2]~~).

rem([H|T], [L, X]~~, D~~):-

Student Name: *Sayed Ahmed Adnan*

Student id: *20025374* Section #: *A*

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

QUIZZ#4: Chapter 16\_Prolog

\*\*\*\*\*

QUESTION: Define a predicate "myflatten" that takes a list INLIST as an argument and returns as a result a list OUTLIST consisting of all elements as INLIST, but with nesting in INLIST removed. INLIST = a list, possibly containing nested lists, OUTLIST = same elements as INLIST, but with nesting removed. Using standard predicates is NOT allowed.

?- myflatten([22, [99,44], [55,[33,88]] ], X).  
X = [22,99,44,55,33,88]

ANSWER:

*INLIST([H|\_])*

*INLIST([H|\_]) :- INLIST(H)*

*myflatten([\_], X)*

*myflatten(A, X) :- ~~INLIST(A)~~ X is INLIST(A),  
myflatten(A, X)*

Student Name: Ilham Ghaz

Student id: 20032850 Section #: A

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

QUIZZ#4: Chapter 16 Prolog

\*\*\*\*\*

QUESTION: Write a Prolog predicate named `takeout` to remove a value (First parameter) from a given list (Second parameter) to produce a new list (Third parameter). If the value is not in a given list, the predicate produces `no`. The next 2 queries may help you.

?- <sup>X</sup>takeout(<sup>Y</sup>5, [a, b, <sup>Z</sup>5, 6, 7], M).

M = [a, b, 6, 7]

Yes

3 ?- takeout(8, [a, b, 5, 6, 7], T).

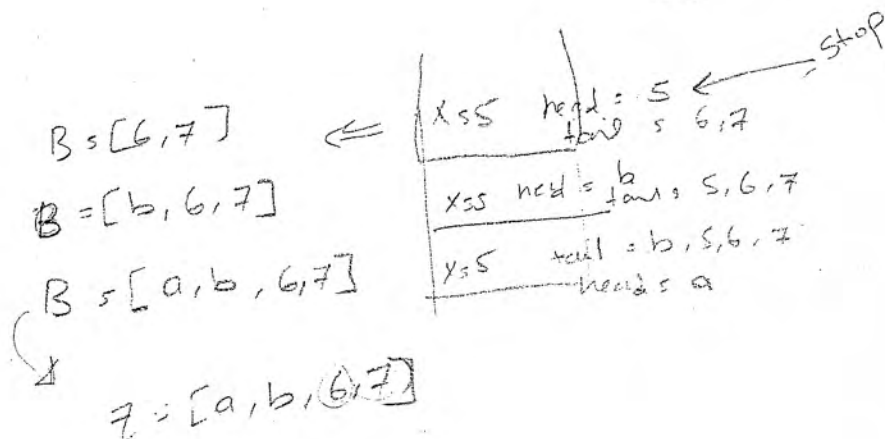
no

ANSWER:

`takeout(X, [X | tail], tail).`

`takeout(X, [head | tail], Z):-`

`takeout(X, tail, B),`  
`append([head], B, Z).`



Student Name: Ali Abdulla

Student id: 20024289 sect #: serial #:

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

QUIZZ#4: Chapter 16\_Prolog

\*\*\*\*\*

QUESTION: Write Prolog rules (predicates) named **double** that takes a list of any numbers and produces a new list each element of which is equal to the double of the corresponding element in the original list.

?- double([3, 2.4, 1.25, 9], U).

U = [6, 4.8, 2.50, 18]

ANSWER:

double([], []).  
double([H|T], [DH|NT]) :-  
DH is H \* 2,  
double(T, NT).

double

pal

do([ ], [ ]).

do([H|T], [R,D]) :-  
1 2 3 4 5      6 7 8

R is H \* 2,

do(T, D).

reverse(L1, L2) :- rev(L1, L2).

rev([ ], L).

rev([X|L], L2, L3) :- rev(L, [X|L2], L3).



Student Name: Ahmed Yousif Ghazwan

Stud id: 2037180

Sect#: 1 #: 2

University of Bahrain

College of Information Technology  
Department of Computer Science

ITCS332: Concepts of Programming Languages

QUIZ#4: Chapter 16 LP theory

\*\*\*\*\*

- 1) In Prolog, the terms in a conjunction are separated by ? ;  
while the terms in a disjunction are separated by ; ✓
- 2) A Proposition is a logical statement that may or may not be true.  
The names used by prolog to store intermediate results start with Small letter
- 3) A Prolog structure that displays the instantiations of values of variables at each step is trace.  
The variable names in by prolog must start with Capital letter
- 4) A Prolog statement consists of terms which may be: Variables,  
or Constant or Structure
- 5) The Prolog query: ?- [t|X] = [H, x, g, b, f] . produces:  
H = t , X = [x, g, b, f]
- 6) The prolog query: ?- [A|B] = [[musa, hud], ahmed, [isa, yusuf]] . produces:  
A = [musa, hud] , and B = [ahmed, [isa, yusuf]]
- 7) The prolog query: ?- [A|B] = [[april, 12, 2010]] . produces:  
A = [april, 12, 2010] , and B = []
- 8) The process of finding a complete sequence of propositions (proof) for the first subgoal before  
working on others is called depth First Search ; the process that works  
on all subgoals of a given goal in parallel is called Breadth First Search
- 9) Prolog operates in 3 modes: Call , exit
- 10) A clausal form of propositions contains Consequent in its left side  
and antecedent in its right side.



Student Name: Mohd Kadhun Abdulla

Student id: 20021538 Section #: A

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

QUIZZ#4: Chapter 16 Prolog

\*\*\*\*\*

- 1) Given the list: Animals = [lion, dog, rabbit, cat, pelican], the head is Lion, the tail is [dog, rabbit, cat, pelican].
- 2) Prolog has a built-in structure named trace that displays the instantiations of values of vars at each step during the attempt to satisfy a given goal.
- 3) The process of finding a complete sequence of propositions (proof) for the first subgoal before working on others is called Depth First search; the process that works on all subgoals of a given goal in parallel is called Breadth First search.
- 4) There are 2 approaches of matching a given goal to a fact in a database: Forward chaining and backward chaining.
- 5) Prolog interpreter says no if the goal cannot be satisfied, and says yes if the goal can be satisfied.
- 6) In a Prolog program, every fact or rule is terminated by .; To continue getting the next answer for a given query, the user types ;.
- 7) The Prolog query: "?- [m|T] = [H, b, f]. " produces:  
H = m, T = [b, f]
- 8) The two sides of a Prolog rule are separated by :-. The terms in a conjunction are separated by ,.
- 9) The right side of rule is called antecedent, and the left side is called consequent.
- 10) The name of a file containing a prolog program is written with an extension .pl.

Student Name: Maitham Ahmed

Student id: 20013738 Section #: 1

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

QUIZZ#4: Chapter 16\_Prolog

\*\*\*\*\*

1) The process of finding a complete sequence of propositions (proof) for the first subgoal before working on others is called Automated; the process that works on all subgoals of a given goal in parallel is called Parallel.

2) There are 2 approaches of matching a given goal to a fact in a database:

rule and functor. The terms in a conjunction are separated by :-.

3) The right side of rule is called conjunction, and the left side is called consequent.

4) The name of a file containing a prolog program is written with an extension .pl.

5) Prolog interpreter says NO if the goal cannot be satisfied, and says Yes if the goal can be satisfied.

6) In a Prolog program, every fact or rule is terminated by .; To continue getting the next answer for a given query, the user types ;.

7) Given the list: Animals=[dog, lion, rabbit, pelican, cat], the head is dog, the tail is [lion, rabbit, pelican, cat].

8) Prolog has a built-in structure named unification that displays the instantiations of values of vars at each step during the attempt to satisfy a given goal.

9) The Prolog query: "?-[t|H]=[T,y,u,r]." produces:

t=T T=T  
H=[y,u,r]  
Yes.  
T=t  
H=[y,u,r]  
Antecedent  
consequent

(1) depth-first / Breadth-first

(2) top-down

(3) :-

(4) Antecedent, consequent

(5) No, yes

(6) Unification

Antecedent  
consequent

Antecedent  
consequent



Student Name: Maithem Ahmed

Student id: 20013738 Section #: 1

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

### QUIZZ#4: Chapter 16\_Prolog

1) The process of finding a complete sequence of propositions (proof) for the first subgoal before working on others is called Automated depth first search, the process that works on all subgoals of a given goal in parallel is called Parallel breadth.

2) There are 2 approaches of matching a given goal to a fact in a database: Backward chaining and Forward chaining.

3) The two sides of a Prolog rule are separated by :. The terms in a conjunction are separated by ^.

4) The right side of rule is called Antecedent, and the left side is called Consequent.

5) The name of a file containing a prolog program is written with an extension .pl.

6) Prolog interpreter says NO if the goal cannot be satisfied, and says Yes if the goal can be satisfied.

7) In a Prolog program, every fact or rule is terminated by .; To continue getting the next answer for a given query, the user types ;.

8) Given the list: Animals=[dog, lion, rabbit, pelican, cat], the head is dog, the tail is [lion, rabbit, pelican, cat].

9) Prolog has a built-in structure named unification table that displays the instantiations of values of vars at each step during the attempt to satisfy a given goal.

10) The Prolog query: "?- [t|H]=[T,y,u,r]." produces:

t=T T=t H=[y,u,r]

H=[y,u,r]

yes.

T=t  
H=[y,u,r]

yes

instantiation



16 Excellent

Student Name: Ali Hassan Alzayez

Student id: 20025557

Section #: A

University of Bahrain  
Department of Computer Science

College of Information Technology  
ITCS332: Concepts of Programming Languages

#### QUIZZ#4: Chapter 16\_Prolog

\*\*\*\*\*

- 1) Given the list: prophets = [ahmed, isa, musa, hud, saleh], the head is ahmed, the tail is [isa, musa, hud, saleh].
- 2) Name 2 types of Prolog statements: rules statement (headed) and fact statement (headed).
- 3) Prolog interpreter says no if the goal cannot be satisfied, and says yes if the goal can be satisfied.
- 4) In a Prolog program, every fact or rule is terminated by ;; To continue getting the next answer for a given query, the user types ;.
- 5) The process of finding a complete sequence of propositions (proof) for the first subgoal before working on others is called depth-first search; the process that works on all subgoals of a given goal in parallel is called breadth-first search.
- 6) There are 2 approaches of matching a given goal to a fact in a database:  
Backward chaining and forward chaining (bottom-up resolution).
- 7) The Prolog query: "?- [m|T] = [H, b, f] ." produces: H = m  
T = [b, f]  
yes
- 8) The two sides of a Prolog rule are separated by :-. The terms in a conjunction are separated by ,.
- 9) The right side of rule is called antecedence, and the left side is called consequent.
- 10) The name of a file containing a prolog program is written with an extension pl.

antecedence

(1) H = [ahmed] T = [isa, ..., saleh]

(2) Query, rule, fact

(10) .pl

(3) No, Yes

(4) dot ,

(5) depth-first search breadth-first search

(6) top-down Bottom-up  
(Backward) (forward)

(7) H = [m] T = [b, f]

(8) :- consequent  
(9) Antecedence conjunction / disjunction